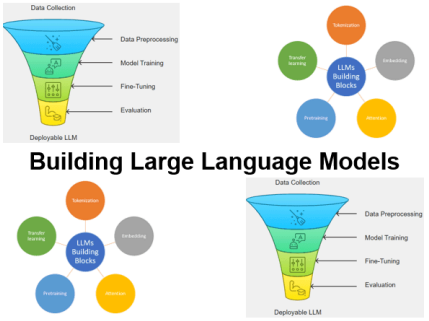
Building Large Language Models

Duration: 4 days **Price: € 3200**

Open Schedule



Audience Course Building Large Language Models

The course Building Large Language Models is intended for engineers who want to design transformer-based LLMs.

Prerequisites Course Building Large Language Models

Participants should be comfortable with Python. Prior exposure to PyTorch or a similar Deep Learning framework is helpful.

Realization Training Building Large Language Models

The training blends concise theory with guided, hands-on labs. Through code-alongs you'll build a mini-GPT, prepare datasets, run pretraining and fine-tuning and deploy models.

Building Large Language Models Certificate

After completion, participants receive a certificate of participation for the course Building Large Language Models.

Content Building Large Language Models

The course Building Large Language Models from SpiralTrain teaches you how to design, train, fine-tune, and deploy transformer-based LLMs using PyTorch and modern tooling. You will learn to build robust text pipelines and tokenizers, implement a GPT-style model with pretraining, apply instruction tuning and RAG and serve models reliably in production.

LLM Intro

The course starts by explaining what LLMs are, where they're used, and the lifecycle of building vs. using them. We introduce the Transformer/GPT architecture, how models learn from large datasets, and when to use classic QA versus RAG.

Working with Text Data

You'll move from raw text to model-ready tensors: tokenization (e.g., BPE), token→ID mapping, special/context tokens, and sliding-window sampling. We cover embeddings and positional encodings while handling unknown words and basic sentence structure.

Attentions Mechanism

This module demystifies self-attention for long-sequence modeling: queries, keys, values, and causal masking to hide future tokens. We add positional encoding, multi-head attention, and stacked layers to capture dependencies across different parts of the input.

Pytorch Deep Learning

This module explains PyTorch fundamentals—tensors, core operations, and training loops—with the tooling to measure model quality. We cover feature scaling/normalization (including categorical features), activation and loss functions, and backpropagation.

Neural Networks

Next the course proceeds with building MLPs and CNNs in PyTorch while choosing appropriate activations and losses and implementing backprop. We touch NLP-specific preprocessing and walk through end-to-end binary and multi-class classification.

GPT from scratch

Next you will implement a minimal GPT with layer normalization, residual (skip) connections, and attention + feed-forward (GELU) blocks with weight tying. You'll also wire up transformer blocks and generate text to see the model in action.

Pretraining

Then pretrain the LLM on unlabeled text with next-token prediction, tracking training vs. validation losses. You will explore decoding strategies (e.g., temperature, top-k), control randomness for reproducibility, and save/load PyTorch weights.

Tuning for Classification

Then the course covers preparing datasets and dataloaders, initializing from pretrained weights, and add a classification head with softmax. Train and evaluate with loss/accuracy, culminating in an LLM-based spam-classification example.

Fine-Tuning

Finally you will practice supervised instruction tuning: format datasets, batch efficiently, and fine-tune a pretrained LLM. Also evaluate outputs, export responses/checkpoints, and apply parameter-efficient methods such as LoRA.

Modules Building Large Language Models

Module 1 : LLM Intro	Module 2 : Working with Text Data	Module 3 : Attentions Mechanism
What is an LLM?	Word Embeddings	Modeling Long Sequences
Applications of LLMs	Decoders and Encoders	Capturing Data Dependencies
Stages of Building LLMs	Decoder Only Transformer	Attention Mechanisms
Stages of Using LLMs	Tokenizing text	Attending Different Input Parts
Transformer Architecture	Convert Tokens into IDs	Using Self-Attention
Utilizing Large Datasets	Special Context Tokens	Trainable Weights
GPT Architecture Internals	Understand Sentence Structure	Hiding Future Words
Learn Language Patterns	Byte Pair Encoding	Positional Encoding
Retrieval Augmented Generation	Unknown Words	Causal Attention
Question and Answer Systems	Sampling with Sliding Window	Masking Weights with Dropout
QA versus RAG	Creating Token Embeddings	Multihead Attention
Building an LLM	Encoding Word Positions	Stacking Attentions Layers
Module 4 : Pytorch Deep Learning	**Module 5 : Neural Networks**	**Module 6 : GPT from scratch**
Deep Learning Intro	Neural Networks Intro	Coding an LLM Architecture
Overview of PyTorch	Building NN with PyTorch	Layer Normalization
PyTorch Tensors	Multiple Layers of Arrays	Normalizing Activations
Tensor Operations	Convolutional Neural Networks	Feed Forward Network
Model Evaluation Metrics	Activation Functions	GELU Activations
Feature Scaling	Loss Functions	Adding Shortcut Connections
Feature Normalization	Backpropagation	Connecting Attention
Categorical Features	Natural Language Processing	Weight Tying
Activation Functions	Stopword Removal	Linear Layers in Transformer Block
Loss Functions	Binary Classification	Coding the GPT Model
Backpropagation	Multi-class Classification	Generating Text
Module 7 : Pretraining	**Module 8 : Tuning for Classification**	**Module 9 : Fine-Tuning**
Pretraining on Unlabeled Data	Categories of Fine-Tuning	Instruction Fine-tuning
Calculating Text Generation Loss	Preparing the Dataset	Supervised Instruction
Training Losses	Creating Data Loaders	Preparing a Dataset
Validation Set Losses	Top-k Sampling	Organizing Training Batches
Training an LLM	Soft-Max Function	Creating Data Loaders
Decoding Strategies	Initializing with Pretrained Weights	Loading a pretrained LLM
Control Randomness	Adding Classification Head	Fine-tuning the LLM
Temperature Scaling	Classification Loss and Accuracy	Extracting and Saving Responses
Saving Model Weights in PyTorch	Fine-tuning on Supervised Data	Evaluating Fine-tuned LLM
Loading Pretrained Weights	Using LLM as Spam Classifier	Fine Tuning with LoRA

SpiralTrain BV
Standerdmolen 10, 2e verdieping
3995 AA Houten

info@spiraltrain.nl
www.spiraltrain.nl
Tel.: +31 (0) 30 – 737 0661

Locations
Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online